



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/786,004	02/26/2004	Russell Norman Owen	13210-16	3548
1059	7590	10/14/2008	EXAMINER	
BERESKIN AND PARR			GUPTA, MUKTESH G	
40 KING STREET WEST				
BOX 401			ART UNIT	PAPER NUMBER
TORONTO, ON M5H 3Y2			2444	
CANADA				
				MAIL DATE
				DELIVERY MODE
				10/14/2008
				PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/786,004	OWEN ET AL.	
	Examiner	Art Unit	
	Muktesh G. Gupta	2444	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 11 August 2008.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-3,6-26 and 29-50 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-3,6-26 and 29-50 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 22 July 2008 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ . |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>08/11/2008 and 06/12/2008</u> . | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| | 6) <input type="checkbox"/> Other: _____ . |

DETAILED ACTION

1. This application has been examined.

Amendments received on 07/22/2008 have been entered.

Claims 1-2, 6, 11, 16, 18-21, 24-25, 29, 32, 36, 39-40, 42, 44-45, 47-48 and 50
are amended.

Claim 4-5 and 27-28 are cancelled.

Claims 1-3, 6-26 and 29-50 have been examined on merits and are pending in
this application.

Information Disclosure Statement

2. Information disclosure statement (IDS) submitted on 06/12/2008 and 08/11/2008
being considered by the examiner and initialed and dated copy is attached to this
office action.

Continued Examination under 37 CFR 1.114

3. A request for continued examination under 37 CFR 1.114, including the fee set
forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since
this application is eligible for continued examination under 37 CFR 1.114, and the
fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous
Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's
submission filed on 07/22/2008 has been entered.

Response to Amendment

4. Acknowledgment is made for Applicants submission for replacement drawings, and claims filed on 07/22/2008 which is entered.

Applicant's arguments with respect to pending claims have been considered but are moot in view of the new ground(s) of rejection.

- a. Applicant's arguments with respect to **Claims 1-3, 6-26 and 29-50** have been considered but are moot in view of the new ground(s) of rejection.
- b. Applicant's arguments and amendments filed on 07/22/2008 have been carefully considered but they are deemed moot in view of the following new grounds of rejection as explained here below, necessitated by Applicant's substantial amendment to the claims which significantly affected the scope thereof.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. **Independent Claims 24 and 50, dependent Claims 25-26 and 29-49** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.
- a. **Claims 24-26 and 29-49** recite the limitation of "A web service accelerator..., which is software per se", and does not have the necessary element of processor or hardware or machine. Since a program which is software per

se, as disclosed on par. 0007, lines 1-2, lacks the necessary physical articles or objects necessary for it to be a machine or a manufacture within the meaning of 35 USC 101, and it's clearly not a series of steps or acts so as to be a process or combination of two or more substances so as to be a composition of matter, it fails to fall within a statutory category. Since the claim is not limited to embodiments eligible for patent protection, it is being rejected as non-statutory as directed to a software per se rather than a patent-eligible machine, manufacture, process or composition of matter. Also see pages 30 and 53 of the Interim Guidelines for Examination of Patent applications for Patent Subject Matter Eligibility.

- b. It is recommended, that **Claims 24** and its dependent be amended as:
“A web service accelerator comprising processor...”.
- c. **Claim 50** recite the limitation of “*A computer-readable medium...*” Applicant intends for the phrase *computer-readable medium* as used in the claims to include “physical or transmission-type media”, where the transmission media includes signals, in addition to anything else which would have reasonably been considered to be a carrier medium by one of ordinary skill in the art, and as stated in par. 0134 line 1-5. As such, the claim covers embodiments directed to signals, per se. Since a signal lacks the necessary physical articles or objects necessary for it to be a machine or a manufacture within the meaning of 35 USC 101, and it's clearly not a series of steps or acts so as to be a process or combination of two or more substances so as to be a

composition of matter, it fails to fall within a statutory category. Since the claim is not limited to embodiments eligible for patent protection, it is being rejected as non-statutory as directed to a form of energy rather than a patent-eligible machine, manufacture, process or composition of matter. **Claim 50** is rejected.

- d. It is recommended, that **Claim 50** be amended as:

"A storage media comprising program instructions which are computer-executable to implement ...".

- e. For the purpose of applying art, **Claims 24-26 and 29-50** are read as with above suggested claim language changes.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. **Claims 1-3, 6-26 and 29-50** are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent Application Publication No. 20050044197 to Lai, Ray Y., (hereinafter "Lai") as applied to **Claims 1-3, 6-26 and 29-50** and further in view of US Patent Application Publication No. 20040148334 to Arellano, Javler B. et al., (herein after "Arellano").

- a. *Regarding **Claims 1-3, 6-26 and 29-50** Lai discloses substantially the invention as claimed. Lai does not explicitly disclose first web service file and creating output file comprising a second web service description file optimized.*
- b. *Lai discloses (as stated in par. 0217, lines 1-11, par. 1426-1434, lines 1-5, par. 1149, lines 1-10, Web Services technology implemented according to embodiments of the generic Web Services architecture may be a good fit. The first exemplary scenario is a membership award program, which is a common customer loyalty program, and provides a seamless integration between a credit-card bonus point system and business partners for redemption, account balance, activity tracking, and cross-selling. The second exemplary scenario is a single payment gateway that enables corporate banking customers to utilize payment services from multiple banks, referred to as a Payment Services scenario. In this example, two Design Patterns are applied: Model-View-Controller Pattern (J2EE Design Pattern) and Service Consolidation-Broker Integration Pattern (B2Bi Pattern). The former denotes that the Control Servlet acts as a Controller (Presentation Tier) for the Model (business logic) and different Views (data sources or remote Web Services). It provides a cleaner approach to decouple the business logic and data from the presentation. The latter denotes a typical Web Services deployment scenario where a Service Broker needs to consolidate and aggregate information from multiple data sources (or remote Web Services). FIG. 111 illustrates the interaction Between Single Sign-on Components according to one embodiment. The components may work together to provide one or more of,*

but not limited to: Login--The Login.jsp accepts the Client's user ID and password for performing Single Sign-on. The Single Sign-on Servlet may generate a one-time session ID and display in the URL (functioning as the parameter for the next servlet). Contents Servlet--The Contents Servlet generates a list of contents URLs (partner Web sites) as the menu page where the Client can choose to access. Forward Servlet--The Forward Servlet forwards the Client to the target partner Web site URL once the SAML Assertion request is processed successfully and sufficient access rights are validated. Ticket Desk--The Ticket Desk handles each SAML Assertion as a request ticket for each partner Web site access. It sends the SAML Assertion request to the Article Servlet of the target Web site for processing. SAML Engine--The SAML APIs that handle SAML Assertion requests and respond to the Client requester. Article Servlet--This module handles forwarding the partner service Web site page to the Client upon successful Single Sign-on. Intermediaries (e.g. information brokers) may offer portal services that consolidate or aggregate different business services. Content or Service Providers may integrate with the portal service, for example via URL rewriting, data exchange using XML and Web Services, or application-to-application interfaces (see FIG. 87). A Partner Directory may be created (e.g. Yellow Pages) to list all business services. UDDI or ebXML Service Registry may be used to implement the Partner Directory. Users (or subscribers) navigate in the Partner Directory and invoke relevant Web Services.

c. *Lai does not disclose that first web service file and creating output file comprising a second web service description file optimized, through it discloses that, SOAP messaging with JAXM is a messaging API that may be used to consolidate and aggregate different information sources to implement the Service Consolidation-Broker Integration Pattern for aggregating, consolidating and optimizing web services.*

d. *Arellano does disclose, (as stated in par. 0021-0031, lines 1-5, agent coordination logic module makes the determination by comparing the ontological description of the web service with the ontological description of the other web service and a method of dynamically aggregating multiple services associated with multiple agents is provided. The method includes implementing a web service with an agent and implementing another web service with another agent. Ontological descriptions of the web service and the other web service are determined according to a shared ontology. Execution of the web service is dynamically coordinated with the other agent according to the determined ontological descriptions. A determination is made whether the web service can be supplemented with the other web service. A determination is made whether the web service and the other web service can be used to provide an improved web service. An improved web service is provided that aggregates the web service and the other web service based on the dynamically coordinated execution of the web service. Dynamic coordination of the execution of the web service automatically occurs independent of a user direction to coordinate*

execution of the web service. Dynamically aggregating multiple services associated with multiple agents. First web service implementing source code segment, that implements a web service and a second web service implementing source code segment that implements another web service. An ontological determination source code segment determines ontological descriptions of the web service and the other web service according to a shared ontology. An execution coordination source code segment dynamically coordinates execution of the web service and the other web service according to the determined ontological descriptions.

e. *It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Lai's Intermediaries (e.g. information brokers) which offer portal web services that consolidate or aggregate different business web services, to Arellano's agent which is supported as a functional module on a device that provides a processor and storage mediums for data and instructions. An agent coordination platform for dynamic aggregation of services is provided with agent coordination logic to coordinate multiple services, e.g., web services that are implemented by at least one agent. The agent coordination platform includes agent coordination logic to dynamically coordinate the agent(s) and services associated with the agent(s). The agent coordination platform also uses a shared ontology so that agent coordination platforms and multiple agents interpret service information according to a common understanding of the services.*

f. *The motivation would have been for an effective and particularly for a way to more efficiently utilize resources by virtue of dynamic aggregation of web services and efficiently providing the requested web services to the client device.*

g. *Therefore, it would have been obvious to combine these two references of Lai's and Arellano's disclosures in light of providing a system, method and program which efficiently integrates multiple web services, optimizes, coordinates and processes at the back end to, provide unified and interactive interface of invoking multiple web services at the client's mobile device.*

*Together Lai and Arellano disclosed all limitations of **Claims 1-3, 6-26 and 29-50** and hence, are rejected under 35 U.S.C. 103(a).*

As to Claims 1, 24 and 50, Lai anticipates method, web services program, stored on computer-readable medium, of processing a web service description so that said web service description is adapted for use with a mobile device, said web service description comprising a plurality of web service description elements, wherein said method is performed at a computing device remotely coupled to the mobile device, said method comprising:

receiving a first web service description file comprising said web service description, wherein said web service description defines an interface to a web service (as stated in par. 0412, lines 1-10, par. 0080, lines 1-2, par. 0217, lines 1-12, Service Requesters (or consumers) may be accessing the business services from a variety of mobile devices or a browser. This belongs to the consumer domain. A client may use a

phone to inquire about an account balance, where all other architecture components are part of the Service Provider domain, where the relevant Web Services components may process the balance inquiry and perform transcoding for different client devices wherever necessary. A client may receive an account balance on a PDA, WAP phone, or another device based on a personalization profile. FIG. 17 illustrates an exemplary Web Services architecture showing the Consumer Domain and Service Provider Domain which is remote to the mobile device and where all the processing of requested web services is done. The following describes two exemplary scenarios where Web Services technology implemented according to generic Web Services architecture. The first exemplary scenario is a membership award program, which is a common customer loyalty program, and provides a seamless integration between a credit-card bonus point system and business partners for redemption, account balance, activity tracking, and cross-selling:

creating at least one accelerator output file from said web service description, said creating comprising optimizing said web service description for said mobile device, wherein said at least one accelerator output file comprises a second web service description file, said second web service description file comprising an optimized web service description is adapted for processing by said mobile device (as stated in par. 0219, lines 1-9, par. 0213, lines 1, par. 0536, lines 6-8, par. 0227, lines 1-8, par. 0167, lines 1-2, par. 1330, lines 1-10, The target candidate for Web Services implementation may be one where existing monolithic services may be decomposed into more atomic business services. By exposing these atomic business services, business information

from various sources may be aggregated with a business process engine into meaningful business information and customer-oriented web services. The second exemplary scenario is a single payment gateway that enables corporate banking customers to utilize payment services from multiple banks, referred to as a Payment Services scenario. The Service Requester is a consumer of business services. This may be the end-user (as in Business-to-Consumer) or server (as in Business-to-Business scenario). The Service Requester finds the business services from the Service Registry via a Service Proxy (such as an Apache SOAP server). Upon a successful search, the Service Registry, which may be provided by the same Service Provider or by a public Service Registry node, fetches the appropriate service description (for example, WSDL) and returns the service end-points (that is where the business service is located) to the Service Requester. *Then the Service Requester can "bind" the business service to the actual service endpoint or location or mobile device.* Dynamic service look-up may be beneficial to managing business services within a large user community, such as millions of content providers and mobile phone subscribers, where business web services and content are typically added or modified in real-time. Web Services technology may be used here to wrap the Point of Sales payment functionality as a reusable business service. This enables the POS or merchandise information (such as payment transactions) to be captured in SOAP messages and reused by the bonus point system. The credit card company may also make available partial contents of the customer transactions with other business service providers who need them to process the award redemption request. This not only

reduces paper work, it can also expedite the processing time of the reward redemption.

FIG. 106 illustrates exemplary Use Cases for an exemplary FX Spot Rate Quote Web Service according to one embodiment. An objective is to use a simplified Foreign Exchange (FX) Spot Rate Quote Service to illustrate how to wrap a remote FX Quote Service using a mixture of XML messaging (XML document using JAXM) and Remote Procedure Call (JAX-RPC) with secure SOAP Message Service (WS-Security) and Security Assertion Markup Language (SAML) protocol for Single Sign-on purpose. How it works is that, given a Sell/Buy currency pair (for example, Sell US Dollars and Buy Euro Dollars), the FX Quote Service will provide a quote based on the average daily exchange rate. Client will receive quote on his mobile device and can execute from his mobile, for buying or selling stock);

and transmitting said at least one accelerator output file to said mobile device, wherein said at least one accelerator output file facilitates invocations of said web service by said mobile device (as stated in par. 0413, lines 1-5, par. 0536, lines 6-8, par. 1340, lines 1-2, par. 1341, lines 1-2, par. 1342, lines 1-8, FIG. 18 illustrates an exemplary detailed Web Services architecture. In the Service Delivery component, of the Service Provider Domain, there is a controller servlet that can handle service requests from the Service Requester's mobile devices or browser. Business data returned from the RPC router servlet may be captured by the service proxy in XML format. The presentation manager may reformat the data and transcoding into HTML, or WML using XSL if applicable. This enables the Service Requester to view in a format that is displayable on any mobile device or browser. The exemplary system preferably

allows users to log in once to access multiple partner services. System preferably validates the user credential against the user access list of the Application Server. In other words, users enter login ID and password at the front page then can access a number of partner Web sites. The exemplary system preferably allows users to enter the Sell and Buy Currency codes to request an indicative FX Spot Rate Quote. This is preferably an online request for a quote, where the exemplary system preferably retrieves the Spot Rate from a remote FX Quote Server);

wherein said optimizing comprises resolving symbolic references in said web service description of said first web service description file such that said at least one accelerator output file is parseable by said mobile device in one pass (as stated in par. 0228, lines 1-7, par. 0546, lines 1-12, par. 0549, lines 1-8, par. 1158, lines 1-12, par. 0585, lines 1-4, par. 0586, lines 1-4, par. 0587, lines 1-4 and par. 0588, lines 1-4, par. 1459, lines 1-3, The bonus point system may also make use of Web Services technology to integrate seamlessly with back-end ERP or legacy systems or to exchange membership award information with service providers via SOAP messages. This allows a flexible and low-cost means of Business-to-Business integration (B2Bi), without creating proprietary and dedicated interfaces. FIG. 26 is a SOAP cache sequence diagram that illustrates some details of the SOAP cache Use Cases previously described. The SOAP client may need to look up a service endpoint URL to invoke a remote business service. The SOAP client checks whether the nature of the remote business service supports cached business service information. An indicative foreign exchange rate inquiry is an example of using SOAP cache. The SOAP client

may then access the SOAP cache, which may be implemented as a hash table stored in the physical memory of the SOAP client. If the cached data is available and is still timely, the SOAP client may retrieve the business service information from the SOAP cache. Predefined transaction types may cache transaction contents and results. Cached data may be stored in serialized data objects, databases, XML cache (edge products), and LDAP with time stamp, version, and transaction type/ID. Cache data (read-only) can be represented in XML structure and can be parsed/retrieved via DOM, SAX, JDOM, or Xpath and output is presented to client mobile device. This Web Services pattern may be useful, for example, for portal integration with multiple Service Providers or content providers doing processing at the back-end. Traditional point-to-point partner integration typically requires complex technical customization for each partner's back-end application infrastructure. Using Web Services technology may preferably lower the cost of partner integration because SOAP messaging may be relatively flexible and less costly to implement. It can also accommodate different data formats and platform connectivity. *Business information can easily be aggregated from multiple sources to create user-stickiness for the portal service with single front-end.* ProfileServlet parses the SOAP request using the JAXM/TSIK Message Provider for optimizing and resolving symbolic references for single sign-on by web service requester using mobile device. Web service description is optimized with administering Multiple Servlet Engines, different configurations for different servlet engines in the server.xml file, different services and resources mapping. Vertical scaling optimizes existing computing capacity by running multiple instances of the SOAP servlet engines.

Multiple servlet engines support application partitioning to preferably provide better performance and scalability);

and wherein said resolving comprises representing the plurality of web service description elements as nodes in a graph, re-ordering the nodes into a tree data structure so that said symbolic references are resolved in a forward direction, and creating said at least one accelerator output file from said tree data structure (as stated in par. 0028-0034, lines 1-5, par. 1484, lines 1-12, par. 0179, lines 1-2, par. 0420-0425, lines 1-11, JAXP is a lightweight Java API library for parsing and transforming XML documents. It is a high-level wrapper for different parsers; it can use Xerces or Crimson as the underlying parser. It allows parsing of an XML document using: Event-driven (SAX 2.0). Tree-based (DOM Level 2). XML documents transformation XML to XML/other data format using XSL/XSLT. Rendering to PDF or graphics using Cocoon.

The exemplary deployment architecture in FIG. 117 shows a possible deployment when the exemplary system is extended and integrated in a typical financial institution's Enterprise Architecture, where there may be different delivery channels supporting, for example, SMS, WAP phone, Fax and email, and interfaces with the external parties, such as the Stock Exchange and Market Data feed providers. The integration of the Web Services components may be handled either by a public or private Service Registry and a SOAP RPC Router. With a Web Services implementation, the XML data transformation may be implemented by JAXP and XSLT, and the Exchange Gateway is implemented using SOAP messaging and/or SOAP-JMS integration. FIG. 118 is a flowchart of a method for generating a vendor-independent Web Service architecture

including a plurality of heterogeneous components in accordance with one or more design patterns according to one embodiment. FIGS. 20 and 21 illustrate exemplary reference architecture for securities trading, with the two Figures showing servers and logical components respectively, according to one embodiment. FIG. 20 illustrates a server-level architecture view of a securities trading (or brokerage) firm that adopts Web Services technology according to one embodiment. The architecture components are categorized into five different tiers based on their functionality or role. Between the tiers, there may be separate routers (thus creating different IP subnets) and firewalls that segregate the servers for security and network management reasons. The Business Tier includes service components that provide the core business logic. In this example, the core online securities trading applications run on clustered J2EE application servers. A private Service Registry (for dynamic service look-up), a set of SOAP servers (acting as a service proxy to back-end legacy systems or remote trading partners' systems), and a market data server (for publishing foreign exchange rates and latest stock prices) also reside in the Business Tier. The Integration Tier hosts the integration components (such as messaging bus), gateways (such as Host Gateway for legacy mainframe systems, and Exchanges gateway for Stock Exchanges), and security components (such as Directory Server and Policy Server). The Host Gateway provides a channel to invoke applications running on legacy mainframes. There is also an Exchanges gateway, which acts as a channel to execute trade orders with local exchanges (such as NASDAQ and JASDAQ) or other markets (such as Instinet, which is an Electronic Communication Network), subscribe market data from market data feeds providers

(such as Reuters or Bloomberg), and clear trade orders with local clearing organizations (such as Hong Kong Exchange, Deposit Trust, and Clearing Corporation). The Directory Server provides enterprise-level authentication. The Policy Server stores access rights and policies that govern the access level of each service component or system by users and by roles. These security components may span two or more tiers. The Resource Tier typically hosts all data stores (such as customer account master and trade data residing on a database server running a relational database), data warehouse, Enterprise Resource Planning (ERP) systems, and legacy mainframe applications. These resources may physically reside on a Storage Area Network (SAN) for better data availability and management. On the client side, the Client Tier may include any client front-end that accesses the online securities trading functionality. This may include, but is not limited to, browsers, rich clients (such as Java SWING client), and mobile devices (such as PDA and WAP phones). The Presentation Tier handles HTTP requests from the client side, processes the presentation logic, and transforms it into some other messaging format. This includes, but is not limited to, the HTTP Web servers (handling static Web pages), a portal server (personalizing contents and aggregating information), and messaging servers (such as SMS server or WAP gateway)).

As to Claims 2, and 25, Lai anticipates method, web services program, of claims 1, and 24, wherein both said web service description of said first web service description file and said optimized web service description of said second web service

description of said second web service description file are in Web Service Description Language (as stated in par. 0262, lines 4-9, Web Services technology uses the Web Services Description Language (WSDL) in the XML structure to define the interfaces, network connection, and service end-points).

As to Claims 3, and 26, Lai anticipates method, web services program, of claims 1, and 24, wherein invocations of the web service by the mobile device are performed in accordance with a Simple Object Access Protocol (as stated in par. 0265, lines 1-4, architecture framework of Web Service is based on open standards technology, J2EETM, XML, SOAP (Simple Object Access Protocol) and UDDI).

As to Claims 6, and 29, Lai anticipates method, web services program, of claims 1, and 24, wherein each of a subset of said plurality of web service description elements is associated with transport protocols not supported by said mobile device, and wherein said optimizing further comprises identifying said subset, and excluding said subset from said at least one accelerator output file (as stated in par. 0972, lines 5-9, Web service technology, middleware integration options use SOAP over HTTPS as the transport which decouples (excludes) the middleware from the data transport integration, making interoperability easier (optimized)).

As to Claims 7, and 30, Lai anticipates method, web services program, of claims 1, and 24, wherein said optimizing further comprises modifying one or more

names associated with each of one or more web service description elements (as stated in par. 0865, lines 1-9, Java transcoding tools allow platform environment parameters, dataset names to be changed (modifying) intelligently to the new target environment).

As to Claims 8, and 31, *Lai anticipates method, web services program, of claims 1, and 24, further comprising validating said at least one accelerator output file* (as stated in par. 0331, lines 1-3 and par. 0332, lines 1-3, Web services technology Compose or transform XML documents, validate their well-formed ness using SAX or DOM with XML Schema and if necessary, transform one format to another format or XML variant or render XML documents into different mobile devices or multi-channel).

As to Claim 9, *Lai anticipates method of claim 1, further comprising processing said at least one accelerator output file by identifying web service description elements that define inputs to said web service, a destination, and a format for said inputs from said optimized web service description* (as stated in par. 0224, lines 1-6, web services technology is easy and quick to deploy and integration framework support different protocols and message formats, including a variety of industry standards and platforms which are identified, integrated and delivered in minimal time frame).

As to Claim 10, *Lai anticipates method of claim 9, further comprising invoking said web service by transmitting input data to said destination in said format* (as stated

in par. 1402, lines 1-16, Open Standards messaging protocols are used for the interaction between different components of web services, Clients, Control Servlet, Reference Data, Service Provider and the Registry Server. Control Servlet handles Presentation-Tier requests, acts as a SOAP client to look up the web services dynamically using JAXR, retrieves reference data using JAXM and invokes remote web services using JAX-RPC).

As to Claim 11, Lai anticipates method of claim 10, further comprising receiving output data from said web service in response to said invoking (as stated in par. 0415, lines 1-6, business data returned from the RPC router servlet is captured by the service proxy in XML format. The presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Web Service Requester to view in a format that is displayable on any mobile device or browser).

As to Claims 12, and 32, Lai anticipates method, web services program, of claims 1, and 24, wherein said optimizing further comprises extracting invocation information from said web service description, and storing said invocation information (as stated in par. 0898, lines 1-11, In this stage, a new CRM business data model (database storage) is defined and customized from the customer information extracted from existing delivery channels such as ATM channel and teller platform. The data extraction, one-way data synchronization using the existing middleware or messaging infrastructure is achieved by implementing web services).

As to Claims 13, and 33, Lai anticipates method, web services program, of claims 12, and 32, further comprising processing said at least one accelerator output file by identifying web service description elements that define inputs to said web service and obtaining operation parameters based on said inputs (as stated in par. 0763, 0764, lines 1-3, front controller servelet initiates EPI, ECI calls, the objective is to invoke a CICS mainframe transaction ID and return the response for terminal operation parameters data).

As to Claims 14, and 34, Lai anticipates method, web services program, of claims 13, and 33, further comprising generating input data by combining said operation parameters with said invocation information (as stated in par. 0777, lines 1-13, CICS resource requests from the CICS socket, web attach program and the alias program CWBA handle the conversion of the code page from ANSI to EBCDIC. The process converts input data from the request to ASCII, which may then be converted to EBCDIC. The analyzer looks up the alias name, converter, user ID, and user program before passing the CICS request to the Alias program CWBA and functions as a converter, which encodes and decodes CICS data).

As to Claims 15, and 35, Lai anticipates method, web services program, of claims 14, and 34, method of claim 14, further comprising invoking said web service by transmitting said input data to said web service (as stated in par. 0777, lines 1-13, Alias

program then invokes the appropriate user application program to process the CICS requests. The Alias program receives the presentation logic decoded in CICS format. The Alias program sends the CICS results in encoded data that can be handled by the presentation logic such as HTML Web pages displayed on the mobile device).

As to Claims 16, and 36, Lai anticipates method, web services program, of claims 15, and 35, further comprising receiving output data from said web service in response to said invoking (as stated in par. 0777, lines 1-13, The Alias program receives the presentation logic decoded in CICS format. The Alias program sends the CICS results in encoded data that can be handled by the presentation logic such as HTML Web pages displayed on the mobile device).

As to Claims 17, and 41, Lai anticipates method, web services program, of claims 1, and 24, wherein said at least one accelerator output file comprises code adapted for execution on said mobile device, for obtaining input data used to invoke said web service, and for invoking said web service using said input data (as stated in par. 0892, lines 1-15 and par. 0729, lines 4-9, J2EE-compliant applications web server provide Web and EJB containers to develop and execute Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX, Java API for XML Pack is a bundle of XML related Java APIs to develop XML based transforming and Web service through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be executed on the mobile platform devices. Web service requester may

start with one single coarse-grained Web service, where different services are invoked by passing parameters in the Web service call either an XML-RPC or an XML message).

As to Claims 18, and 42, Lai anticipates method, web services program, of claims 17, and 41, wherein said creating step comprises:

identifying web service description elements that define inputs to said web service from said web service description (as stated in par. 0301, lines 1-7, web service technologies implement service proxy, JAXR, search engines for searching registry with their search algorithm, combination of logical operators for identifying elements that define inputs to web service);

producing first instructions for generating a user interface to prompt a user for one or more of said inputs to said web service (as stated in par. 0302, lines 1-7 and par. 0310, lines 1-7JAXR pluggable provider class is able to use any capability-specific interfaces such as ebXML provider or UDDI provider that is specific to a particular Service Registry platform and require to input to many data entry screens for creating a service record and access registry content, objects, catalogs, specifications, WSDL documents, standards and XML schema);

producing second instructions for obtaining input data associated with said one or more inputs (as stated in par. 0303, lines 1-7, APIs or servlets are used to access their registry contents. When Java API for Registries JAXR is used, the same Java API

may be used with different registry properties. Property file stores the specific registry servlet or URIs, Universal Resource Identifier, or the service endpoints URLs);

identifying web service description elements that define a destination and a format for said inputs to said web service (as stated in par. 0240, lines 1-7, web service providers use different messaging protocols and data formats, they use SOAP or ebXML messaging to exchange documents or instructions. The benefit of using SOAP or ebXML messaging is that they are able to integrate with the client or service providers' back-end systems. Documents encapsulated in XML structure within a SOAP message are transcoded into a format that can be understood by the back-end ERP or legacy systems);

and producing third instructions for invoking said web service by transmitting said input data to said destination in said format (as stated in par. 0422, lines 1-7, controller servlet are documents or instructions encapsulated in XML structure for the Integration Tier to hosts the integration components such as messaging bus, gateways such as Host Gateway for legacy mainframe systems, and security components such as Directory Server and Policy Server. The Host Gateway provides a channel to invoke applications running on legacy mainframes for appropriate web services).

As to Claims 19, and 44, Lai anticipates method, web services program, of claims 18, and 43, wherein said creating step further comprises:

identifying web service description elements that define outputs from said web service in response to invocations of said web service and a format for said outputs

from said web service description (as stated in par. 0422, lines 1-18, controller servlet are documents or instructions encapsulated in XML structure for the Integration Tier to hosts the integration components such as messaging bus, gateways such as Host Gateway for other web services. The Directory Server provides enterprise-level authentication for web services. The Policy Server stores access rights and policies that govern the access level of each web service component or system by users and by roles. These security components span two or more tiers);

and producing fourth instructions for receiving output data in said format from said web service (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, messaging calls and data returned from the RPC router servlet is captured by the service proxy from the Resource Tier which hosts all data stores residing on a database server running a relational database, data warehouse, Enterprise Resource Planning (ERP) systems, and legacy mainframe applications).

As to Claims 20, and 45, Lai anticipates method, web services program, of claims 19, and 44, wherein said creating step further comprises:

producing fifth instructions for outputting output data received from said web service to said user (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Service Requester (user) to view in a format that is displayable on any mobile device or browser).

As to Claims 21, and 48, Lai anticipates method, web services program, of claims 17, and 35, wherein said creating further comprises compiling instructions produced at said creating into said code (as stated in par. 0872, lines 1-6, web services technologies implement cross-compiling of the source program language structure such as COBOL to a target program language structure such as Java byte-code using an intelligent language cross-compiler without changing the application program logic, structured procedures and instructions may then be transcoded into Java beans or EJBs for optimizing web services).

As to Claims 22, and 49, Lai anticipates method, web services program, of claims 21, and 35, wherein said code represents an executable Java application (as stated in par. 0872, lines 1-6, web services technologies implement cross-compiling of the source program language structure such as COBOL to a target program language structure such as Java byte-code using an intelligent language cross-compiler without changing the application program logic, structured procedures and instructions may then be transcoded into Java beans or EJBs for optimizing web services).

As to Claim 23, Lai anticipates method, of claim 17, further comprising processing said at least one accelerator output file by executing said code (as stated in par. 0892, lines 1-15 and par. 0729, lines 4-9, J2EE-compliant applications web server provides Web and EJB containers to develop and execute Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX,

Java API for XML Pack is a bundle of XML-related Java APIs to develop XML -based transforming and Web services through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be executed on the mobile platform devices. Web services requester may start with one single coarse-grained Web service, where different services are invoked by passing parameters in the Web service call either an XML-RPC or an XML message).

As to Claim 37, Lai anticipates, web services program of claim 36, wherein said method further comprises transmitting at least a subset of said output data to said mobile device (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Service Requester (user) to view in a format that is displayable on any mobile device or browser).

As to Claim 38, Lai anticipates, web services program of claim 24, wherein said method further comprises receiving input data from said mobile device and invoking said web service by transmitting said input data to said web service (as stated in par. 0967, lines 1-19, par. 0892, lines 1-15 and par. 0729, lines 4-9, client requests for the web services start with inputting data as User ID and password, are represented in SOAP messages, which are carried over HTTPS. SOAP messages may also be carried over other data transport, such as SMTP or FTP. J2EE-compliant applications web server provides Web and EJB containers to develop and execute Java servlets and

EJBs. It also supports session, state, and connection pooling for transaction processing. JAX, Java API for XML Pack is a bundle of XML -related Java APIs to develop XML -based transforming and web services through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be executed on the mobile platform devices. Web services requester may start with one single coarse-grained web service, where different services are invoked by passing parameters in the web services call either an XML -RPC or an XML message).

As to Claim 39, Lai anticipates, web services program of claim 38, wherein said method further comprises receiving output data from said web service in response to said invoking step and transmitting said output data to said mobile device (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Service Requester (user) to view in a format that is displayable on any mobile device or browser).

As to Claim 40, Lai anticipates, web services program of claim 39, wherein said method further comprises detecting changes to said output data from said web service in response to said invoking and transmitting said changes to said mobile device (as stated in par. 0240, lines 1-7, web service providers use different messaging protocols and data formats, they use SOAP or ebXML messaging to exchange documents or instructions. The benefit of using SOAP or ebXML messaging is that they are able to

integrate with the client or service providers' back-end systems. Documents encapsulated in XML structure within a SOAP message are transcoded into a format that can be understood by the back-end ERP or legacy systems. Thus, the integration effort may be lower and reusable for other web services, and may not require all web service providers to use the same vendor solution or to adopt a proprietary data format. This enables any changes from back-end systems to be transmitted back to Service Requester (user) to view in a format that is displayable on any mobile device or browser).

As to Claim 43, Lai anticipates, web services program of claim 42, wherein said third instructions comprise instructions for receiving said input data from said mobile device and transmitting said input data to said web service (as stated in par. 0967, lines 1-19, par. 0892, lines 1-15 and par. 0729, lines 4-9, client requests for the web services start with inputting data as User ID and password, are represented in SOAP messages, which are carried over HTTPS. SOAP messages may also be carried over other data transport, such as SMTP or FTP. J2EE-compliant applications web server provides Web and EJB containers to develop and execute Java servlets and EJBs. It also supports session, state, and connection pooling for transaction processing. JAX, Java API for XML Pack is a bundle of XML-related Java APIs to develop XML-based transforming and web services through JAXP, JAXB, JAXM, JAXR, and JAX-RPC modules to be executed on the mobile platform devices. Web services requester may start with one single coarse-grained web service, where different services are invoked

by passing parameters in the web services call either an XML -RPC or an XML message).

As to Claim 46, Lai anticipates, web services program of claim 45, wherein said fifth instructions comprise instructions for receiving said output data from said web services and for transmitting said output data to said mobile device (as stated in par. 0423, lines 1-8 and par. 0415, lines 1-5, presentation manager reformats the data and transcode into HTML, or WML using XSL if applicable. This enables the Service Requester (user) to view in a format that is displayable on any mobile device or browser).

As to Claim 47, Lai anticipates, web services program of claim 46, wherein said method further comprises the step of detecting changes to said output data from said web service in response to said invoking step and transmitting said changes to said mobile device (as stated in par. 0240, lines 1-7, web service providers use different messaging protocols and data formats, they use SOAP or ebXML messaging to exchange documents or instructions. The benefit of using SOAP or ebXML messaging is that they are able to integrate with the client or service providers' back-end systems. Documents encapsulated in XML structure within a SOAP message are transcoded into a format that can be understood by the back-end ERP or legacy systems. Thus, the integration effort may be lower and reusable for other web services, and may not require all web service providers to use the same vendor solution or to adopt a

Art Unit: 2144

proprietary data format. This enables any changes from back-end systems to be transmitted back to Service Requester (user) to view in a format that is displayable on any mobile device or browser).

Remarks

7. The following pertaining arts are discovered and not used in this office action.
Office reserves the right to use these arts in later actions.
 - a. Connor; Kathy R. et al. (US 7349949 B1) System and method for facilitating development of a customizable portlet
 - b. Carson, Carollyn et al. (US 20040093580 A1) System and methodology for mobile e-services
 - c. Katz, Neil et al. (US 20050071448 A1) Mobile application and content provisioning using web services technology
 - d. Perham, Michael et al. (US 20050086330 A1) Method and apparatus for dynamic, non-intrusive personalization of web services
 - e. Shenfield, Michael et al. (US 20040199614 A1) Information processing system, information disclosing server, and portal server

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Muktesh G. Gupta whose telephone number is 571-270-

5011. The examiner can normally be reached on Monday-Friday, 8:00 a.m. -5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William C. Vaughn can be reached on 571-272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MG

/William C. Vaughn, Jr./

Supervisory Patent Examiner, Art Unit 2444